

VME Reverse Engineering

Originalartikel

Backup

<html> <p>With some free time on his hands waiting for delayed parts to arrive, [Rik] set out to <a href=„<https://justanotherelectronicsblog.com/?p=876>“ target=„_blank“>reverse engineer an old VME system he had acquired. <a href=„<https://en.wikipedia.org/wiki/VMEbus>“ target=„_blank“>VMEbus computers are based on the standard Eurocard PCB format, which defines a wide range of card sizes — the most common being 6U height like [Rik]’s system. They usually consist of a rack-mounted card cage with a passive backplane. Originally, Motorola 68000-based CPU cards were used in VMEbus systems, but any processor could be used as long as you provided the right signals and timings to the system bus. Eurocard systems are less common these days, but are still used in some applications. In fact, if you’re into synthesizers, you may be using Eurocards today — the <a href=„<https://en.wikipedia.org/wiki/Eurorack>“ target=„_blank“>Eurorack standard is based on the standard 3U card size.</p><p>Back to [Rik]’s project, he had no idea what this system was nor how to use it. A bit of probing around and he found two UARTs, a system monitor, and a way to load and dump S-record files. He documents the process quite well, as the internal layout and memory map of the system is unlocked piece by piece. We also like his method of instrumenting the VMEbus signals — logic analyzers are so small today, you can just mount one inside the rack.</p><p><img data-attachment-id=„475240“ data-permalink=„<https://hackaday.com/2021/05/05/vme-reverse-engineering/vme-logic-analyzer/>“ data-orig-file=„<https://hackaday.com/wp-content/uploads/2021/05/vme-logic-analyzer.png>“ data-orig-size=„1213,741“ data-comments-opened=„1“ data-image-meta=„{“aperture”:0,“credit”:“”,“camera”:“”,“caption”:“”,“created_timestamp”:0,“copyright”:“”,“focal_length”:0,“iso”:0,“shutter_speed”:0,“title”:“”,“orientation”:0,“}:“ data-image-title=„vme-logic-analyzer“ data-image-description=„<p>image from project blog</p>“ data-image-caption=„“ data-medium-file=„<https://hackaday.com/wp-content/uploads/2021/05/vme-logic-analyzer.png?w=400>“ data-large-file=„<https://hackaday.com/wp-content/uploads/2021/05/vme-logic-analyzer.png?w=800>“ class=„alignnone size-large wp-image-475240“ src=„<https://hackaday.com/wp-content/uploads/2021/05/vme-logic-analyzer.png?w=800>“ alt=„“ width=„800“ height=„489“ srcset=„<https://hackaday.com/wp-content/uploads/2021/05/vme-logic-analyzer.png> 1213w, <https://hackaday.com/wp-content/uploads/2021/05/vme-logic-analyzer.png?resize=250,153> 250w, <https://hackaday.com/wp-content/uploads/2021/05/vme-logic-analyzer.png?resize=400,244> 400w, <https://hackaday.com/wp-content/uploads/2021/05/vme-logic-analyzer.png?resize=800,489> 800w“ referrerpolicy=„no-referrer“ /></p><p>Spoiler alert: [Rik] succeeds in mapping out the memory, writes some small programs in 68k assembly language, and even builds his own LED accessory card so he can blink some lights (as one must do).</p><p>We <a href=„<https://hackaday.com/2021/04/27/ask-hackaday-why-make-modular-hardware/>“>wrote about modularity recently, and VMEbus + Eurocard systems are good examples of modular design. You could quickly put together a robust assembly using entirely off-the-shelf cards, or mix in your own custom cards. But technology advancements in clock speeds and miniaturization have made these card cage, passive backplane systems less and less relevant today. Do any of you still use the

VMEbus, or have you designed with them in the past? Let us know down in the comments below.</p></html>

From:
<https://schnipsl.qgelm.de/> - **Qgelm**

Permanent link:
<https://schnipsl.qgelm.de/doku.php?id=wallabag:wb2vme-reverse-engineering>

Last update: **2025/06/27 11:17**

